



*Univerzitet Singidunum Beograd,
Tehnički fakultet*

Internet stvari

Izveštaj

Simulacija ponašanja semafora (Arduino Uno)

Student:
Đorđe Krstović

Beograd, 2020

Sadržaj

Sadržaj	2
Uvod	3
Komponente	4
Šema povezivanja kola.....	5
Dijagram stanja	6
Idejno rešenje	7
Kod implementacije.....	8
Arduino 1 – kod implementacije:	8
Arduino 2 – kod implementacije:	12
Link do projekta na tinkercad platformi	13

Uvod

Simulacija ponašanja semafora (SPS) - oponaša rad semafora za vozila i pešake u saobraćaju. Sistem simulira ponašanje semafora u tačno određenim vremenskim intervalima. SPS predstavlja sistem od dva semafora, semafor za pešake (crveno/zeleno svetlo) sa tasterom, i semafor za vozila (crveno/žuto/zeleno svetlo). Za simulaciju ovog sistema potreban je "online" softver Tinkercad, Arduino Uno mikrokontrolerska ploča otvorenog koda, uz korišćenje LED dioda, prekidača, i otpornika.

Komponente

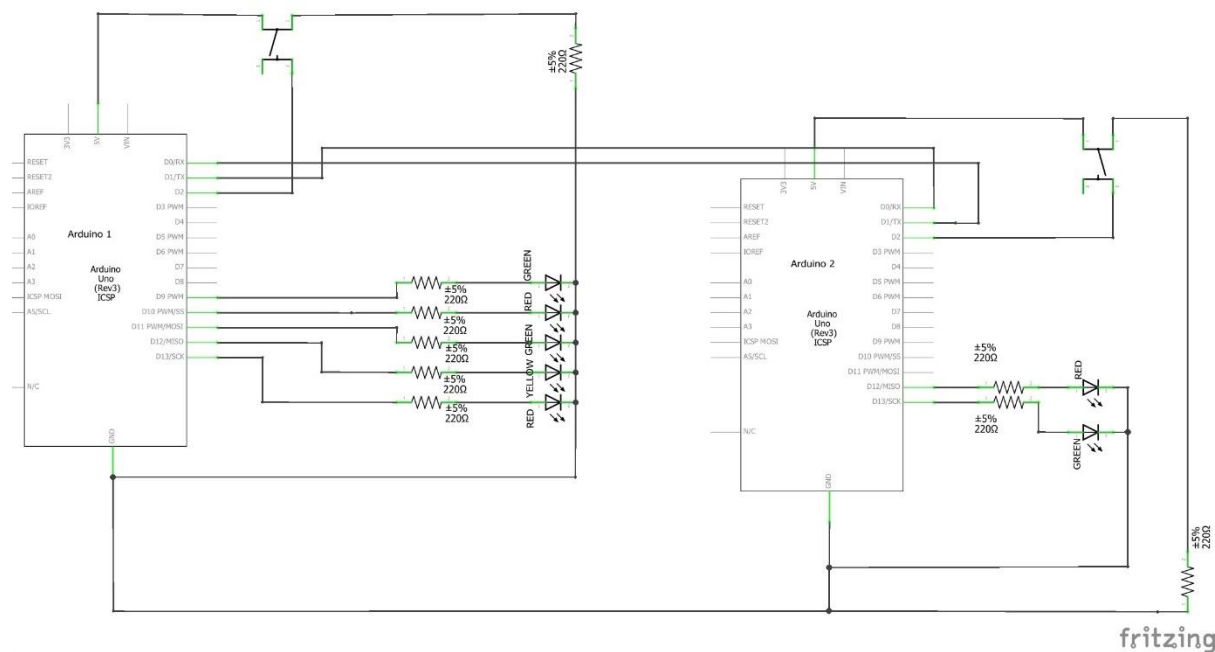
Tabelarni prikaz svih komponenti koje su potrebne za realizaciju ovog simulatora, uz podatke o količini i napomenama koje šire opisuju komponente kao i njihovu namenu.

Komponenta	Količina	Napomena
Arduino Uno	2	/
Red LED	3	1 za vozila, 1 za pešake, 1 za nadsistem
Yellow LED	1	/
Green LED	3	1 za vozila, 1 za pešake, 1 za nadsistem
Resistor	10	220 Ω svi
Pushbutton	2	/

Komponente - tabela

Šema povezivanja kola

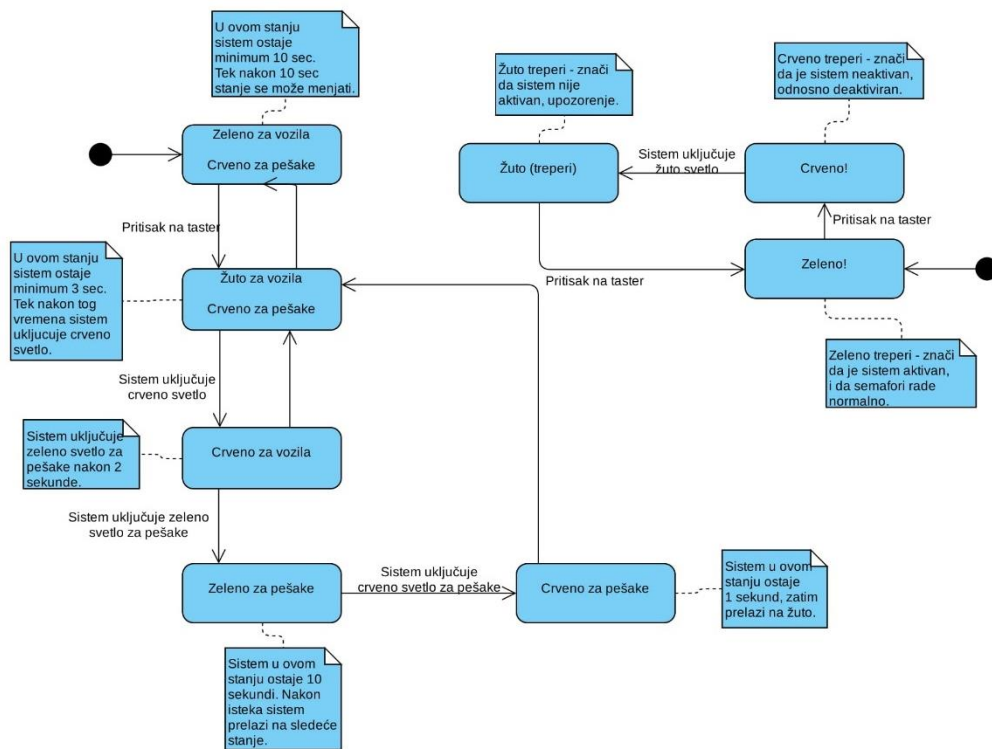
Šema povezivanja kola za simulator ponašanja semafora realizovana u softveru Fritzing korišćenjem svih navedenih komponenti u tabeli.



Šema povezivanja kola

Dijagram stanja

Dijagram stanja koji ilustruje ideju realizacije programa uz sva stanja i napomene. Sa leve strane prikazana su stanja za Arduino1 koji simulira rad semafora za vozila i pešake, a sa desne strane se nalaze stanja za Arduino2 koji simulira rad nadsistema koji kontroliše semafor sa leve strane.

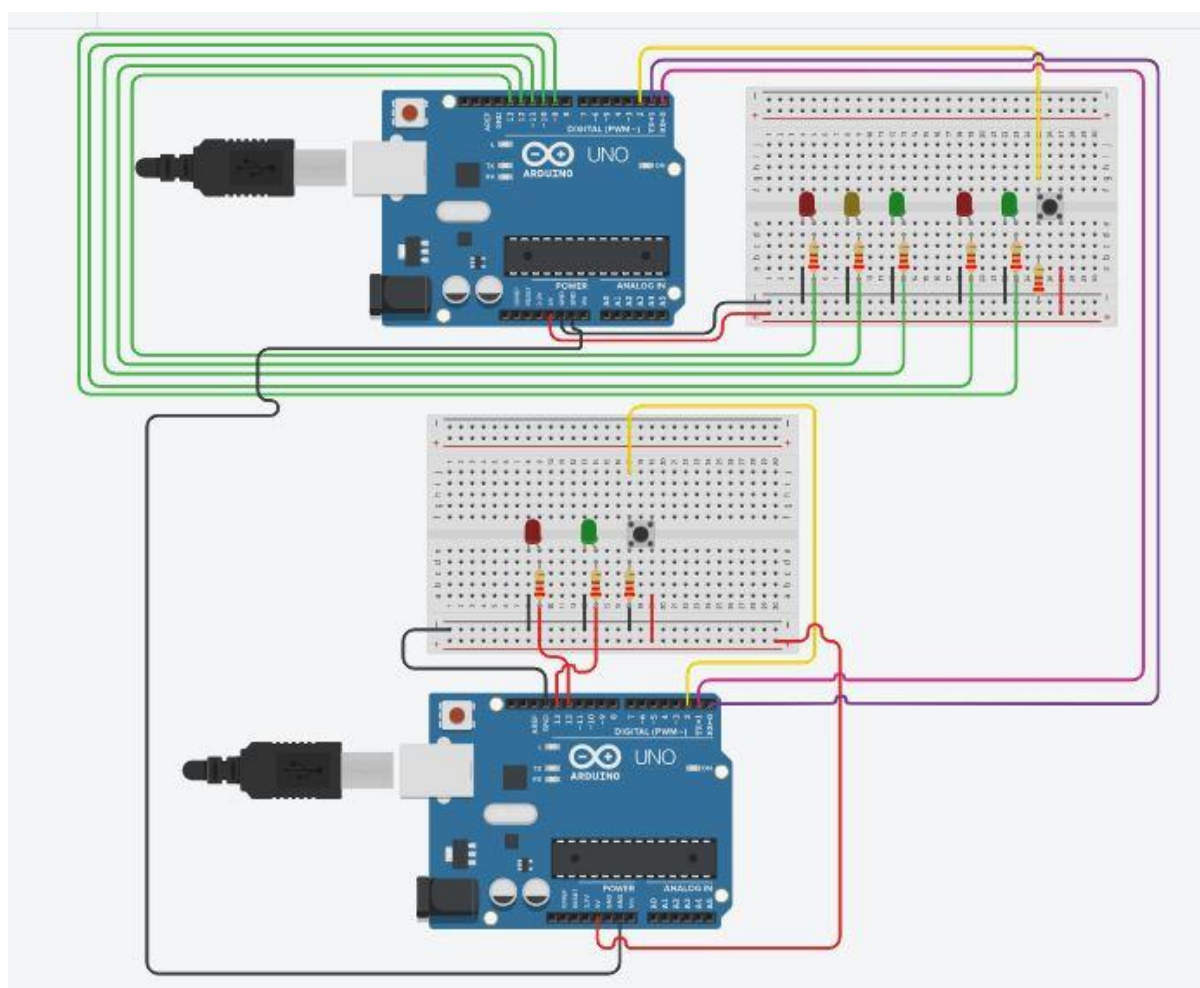


Dijagram stanja

Idejno rešenje

Realizacija simulacije ponašanja semafora korišćenjem Arduino uno mikrokontrolerske ploče i komponenti koje su navedene u tabeli. Gde svaka komponenta predstavlja jednu komponentu u semaforima koji se nalaze u saobraćaju. Sve komponente predstavljaju 1:1 simulaciju, pa npr. crvena, zelena, i žuta LED dioda za vozače predstavljaju zapravo sva tri stanja na semaforu u saobraćaju. CRVENO – stop, ŽUTO – upozorenje, ZELENO – kreni. Uz to imamo i semafor za pešake sa tasterom, koji ima samo 2 svetla, crveno i zeleno, takođe 1:1 simulacija, CRVENO – čekaj, ZELENO – kreni stanje se menja pritiskom na taster. Svako od stanja ima svoje vremensko ograničenje, odnosno koliko dugo će se to stanje zadržati. Takođe ova simulacija poseduje i nadсистem za kontrolu semafora, npr. kada trepti zelena LED dioda, to znači da sistem radi kako treba, ukoliko je taster pritisnut to simulira neispravnost semafora i u ovom slučaju žuta LED dioda treperi.

Prikaz simulacije komponenti i povezivanja – Tinkercad:



Tinkercad – Simulacija ponašanja semafora

Kod implementacije

Arduino 1 – kod implementacije:

```
1.  int flag;
2.  int state = 1;
3.  int buttonState = 0;
4.  String message;
5.
6.
7.  // kad sistem ne radi, odnosno treperi žuto
8.  // svetlo, poziva se promenom stanja semafora
9.
10. void yellowBlinking(){
11.
12.  // sve diode setujemo na LOW da ne bi
13.  // svetlele dok žuto svetlo treperi
14.
15.    digitalWrite(13, LOW);
16.    digitalWrite(11, LOW);
17.    digitalWrite(10, LOW);
18.    digitalWrite(9, LOW);
19.
20.  // dok god je stanje jednako 0, žuto svetlo
21.  // treperi i signalizira da nešto nije u redu
22.
23.    while(state == 0){
24.
25.      // pin 12 predstavlja žutu LED diodu i
26.      // parametrima HIGH i LOW definišemo, da li svetli
27.      // ili je ugašena, odnosno simuliramo treperenje
28.
29.      digitalWrite(12, HIGH);
30.      delay(1000); // čeka 1 sekund
31.      digitalWrite(12, LOW);
32.      delay(1000); // čeka 1 sekund
33.      change();
34.    }
35.  }
36.
37.
38.
39.  // inicijalno stanje pri pokretanju simulacije
40.
41.  void initState(){
42.
43.    //Zeleno svetlo za vozila, crveno za pešake,
44.    // odnosno opet definišemo pomoću HIGH i LOW,
45.    // koja dioda svetli a koje su ugašene.
46.
47.    digitalWrite(13, LOW);
48.    digitalWrite(12, LOW);
49.    digitalWrite(11, HIGH);
50.
51.    // u ovom slučaju kada je zeleno za vozila (kod gore)
52.    // definišemo da je crveno za pešake (kod dole)
53.
54.    digitalWrite(10, HIGH);
55.    digitalWrite(9, LOW);
56.    delay(10000); // čeka 10 sekundi
57.  }
58.
59.
60.
```



```

61. // promena stanja semafora, i ukoliko je "Off" žuto treperi
62.
63. void change(){
64.     if(Serial.available()>0){
65.         message = Serial.readString();
66.         Serial.print(message);
67.         if (message == "On "){
68.
69.             state = 1;
70.             loop();
71.
72.         }
73.         if (message == "Off"){
74.
75.             state = 0;
76.             yellowBlinking();
77.
78.         }
79.     }
80. }
81.
82.
83.
84. void setup()
85. {
86.     pinMode(2, INPUT);
87.
88.     pinMode(13, OUTPUT);
89.
90.     pinMode(12, OUTPUT);
91.
92.     pinMode(11, OUTPUT);
93.
94.     pinMode(10, OUTPUT);
95.
96.     pinMode(9, OUTPUT);
97.
98.     pinMode(3, INPUT);
99.
100.     Serial.begin(9600);
101. }
102.
103.
104.
105. void loop()
106. {
107.     change();
108.     flag = 0;
109.
110.     // Zeleno svetlo za vozila, crveno svetlo za pešake
111.     // takođe korišćenjem HIGH i LOW
112.
113.     digitalWrite(13, LOW);
114.     digitalWrite(12, LOW);
115.     digitalWrite(11, HIGH);
116.     digitalWrite(10, HIGH);
117.     digitalWrite(9, LOW);
118.
119.     delay(10000); // čeka 10 sekundi
120.
121.     change();
122.
123.     bool btnFlag = false;
124.
125.     while(flag == 0){
126.         buttonState = digitalRead(2);

```

```
127.
128.
129.     change();
130.
131.     if(buttonState==HIGH){
132.         btnFlag=true;
133.     }
134.
135.     if(buttonState==LOW && btnFlag==true){
136.         btnFlag = false;
137.         //Zuto svetlo za vozila, crveno svetlo za pešake
138.
139.         digitalWrite(13, LOW);
140.         digitalWrite(12, HIGH);
141.         digitalWrite(11, LOW);
142.         digitalWrite(10, HIGH);
143.         digitalWrite(9, LOW);
144.
145.         delay(3000); // čeka 3 sekunde
146.
147.
148.
149.
150.         // pa se izvrši promena
151.         change();
152.
153.         //Crveno svetlo za vozila, crveno svetlo za pešake
154.
155.         digitalWrite(13, HIGH);
156.         digitalWrite(12, LOW);
157.         digitalWrite(11, LOW);
158.         digitalWrite(10, HIGH);
159.         digitalWrite(9, LOW);
160.
161.         delay(2000); // čeka 2 sekunde
162.
163.
164.         // opet se izvrši promena
165.         change();
166.
167.         //Crveno svetlo za vozila, zeleno svetlo za pešake
168.
169.         digitalWrite(13, HIGH);
170.         digitalWrite(12, LOW);
171.         digitalWrite(11, LOW);
172.         digitalWrite(10, LOW);
173.         digitalWrite(9, HIGH);
174.
175.         delay(10000); // čeka 10 sekundi
176.
177.
178.
179.         change();
180.
181.         //Crveno svetlo za vozila, crveno svetlo za pešake
182.
183.         digitalWrite(13, HIGH);
184.         digitalWrite(12, LOW);
185.         digitalWrite(11, LOW);
186.         digitalWrite(10, HIGH);
187.         digitalWrite(9, LOW);
188.
189.         delay(1000); // čeka 1 sekundu
190.
191.         change();
192.
```

```
193.          //Zuto svetlo za vozila, crveno svetlo za pešake
194.
195.          digitalWrite(13, LOW);
196.          digitalWrite(12, HIGH);
197.          digitalWrite(11, LOW);
198.          digitalWrite(10, HIGH);
199.          digitalWrite(9, LOW);
200.
201.          delay(3000); // čeka 3 sekunde
202.
203.
204.
205.          change();
206.
207.          flag = 1;
208.
209.          }
210.      }
211.  }
```

Arduino 2 – kod implementacije:

```
1. int buttonState = 0;
2. int state = 1;
3. int sState;
4.
5.
6.
7. void setup(){
8.
9. // definicija pinova, da li su input ili output
10.
11. pinMode(2, INPUT);
12. pinMode(13, OUTPUT);
13. pinMode(12, OUTPUT);
14.
15. Serial.begin(9600);
16.
17. attachInterrupt(digitalPinToInterrupt(2), change, FALLING);
18. }
19.
20.
21.
22. void loop(){
23.
24. // stanja simulacije, radi ili ne radi
25.
26. if(state == 1){
27.
28.     working();
29.
30. }
31.
32. if(state == 0){
33.
34.     NWorking();
35.
36. }
37. }
38.
39.
40.
41. // ako radi zeleno treperi na svaki sekund
42. void working(){
43.
44.     state = 1;
45.     sState = 0;
46.
47.     digitalWrite(13, HIGH);
48.     delay(1000); // čeka 1 sekundu
49.
50.     digitalWrite(13, LOW);
51.     delay(1000); // čeka 1 sekundu
52. }
53.
54.
55. // ako ne radi crveno treperi na svaki sekund
56. void NWorking(){
57.
58.     state = 0;
59.     sState = 1;
60.
61.     digitalWrite(12, HIGH);
```

```
62. delay(1000); // čeka 1 sekundu
63.
64. digitalWrite(12, LOW);
65. delay(1000); // čeka 1 sekundu
66. }
67.
68.
69.
70. void change(){
71.   state = sState;
72.
73.   if (state == 1){
74.     Serial.write("On ",3);
75.     delay(1000); // čeka 1 sekundu
76.
77.   }
78.   if (state == 0){
79.     Serial.write("Off",3);
80.     delay(1000); // čeka 1 sekundu
81.
82.   }
83. }
84. }
85. }
```

Link do projekta na tinkercad platformi

<https://www.tinkercad.com/things/d4QExN6A4cV-semafori-klk1/editel?sharecode=9tp8Z5sPdiy6gJKMvgFfGkGTOET8rZsZrvn05-V4ZFE>